Introduction to Privacy Preserving Distributed Data Mining

Murat Kantarcioglu

Privacy vs. Data Mining ???

- **“Both Parties Wary of Data Mining”**
  - (Wired, 01.25.03)
- **“Panel Urges New Protection On Federal Data Mining“**
  - (NYT, 05.17.04)
- **“Survey Finds U.S. Agencies Engaged in 'Data Mining’ “**
  - (NYT, 05.27.04)
- PRIVACY CAN BE PRESERVED!!!

Privacy and Security Restrictions

- Individual Privacy
  - Nobody should know more about any entity after the data mining than they did before
- Organization Privacy
  - Protect knowledge about a collection of entities
    - Individual entity values may be known to all parties
    - Which entities are at which site may be secret

Privacy constraints don't prevent data mining

- Goal of data mining is summary results
  - Association rules
  - Classifiers
  - Clusters
- The results alone need not violate privacy
  - Contain no individually identifiable values
  - Reflect overall results, not individual organizations

The problem is computing the results without ***disclosing*** *the data!*

Privacy-Preserving Distributed Data Mining: Why ?

- Data needed for data mining maybe distributed among parties
  - Credit card fraud data
- Inability to share data due to privacy reasons
  - HIPPAA
- Even partial results may need to be kept private

Secure Multi-Party Computation (SMC)

- The goal is computing a function $f(x_1, x_2, \cdots, x_n)$ without revealing x_i
- Semi-Honest Model
  - Parties follow the protocol
- Malicious Model
  - Parties may or may not follow the protocol
- We cannot do better then the existence of the third trusted party situation
- Generic SMC is too inefficient for PPDDM

Secure Multiparty Computation: Definitions

- Secure
  - Nobody knows anything but their own input and the results
  - Formally: $\exists$ polynomial time S such that $\{S(x,f(x,y))\} \equiv \{View(x,y)\}$
- Semi-Honest model: follow protocol, but remember intermediate exchanges
- Malicious: “cheat” to find something out

Distributed Association Rule Mining: Definitions

- Assume there are n sites with transaction databases $D_1, D_2, \ldots\ldots D_n$ where each has size $|D_i|$
- An itemset X has a local support $X.\sup_i$
- The global support for X (X.sup)

$$X.\sup = \sum_{i=1}^{n} X.\sup_i$$

Definitions: Continues..

- $X \Rightarrow Y$ is globally supported if

$$\{X \cup Y\}.sup \geq s * \sum_{i=1}^{n} |DB_i|$$

- Global confidence of rule $X \Rightarrow Y$ is
  $\{X \cup Y\}.sup / X.sup$
- *Distributed association rule mining*
  - Rules of the form $X \Rightarrow Y$ that has global support and confidence above certain thresholds

Privacy-Preserving Distributed Association Rule Mining.

- Exchanging support counts is enough for mining association rules
- We do not want to reveal
  - which rule is supported(or not) at which site
  - the support count of each rule
  - the database sizes
  - *e.g.* Hospitals may not want to reveal procedures with high mortality rates
  - e.g. Companies may not want to reveal the traces of intrusions

Overview of the Method

1. Find the union of the locally large candidate itemsets securely

2. After the local pruning, compute the globally supported large itemsets securely

3. Check the confidence of the potential rules securely

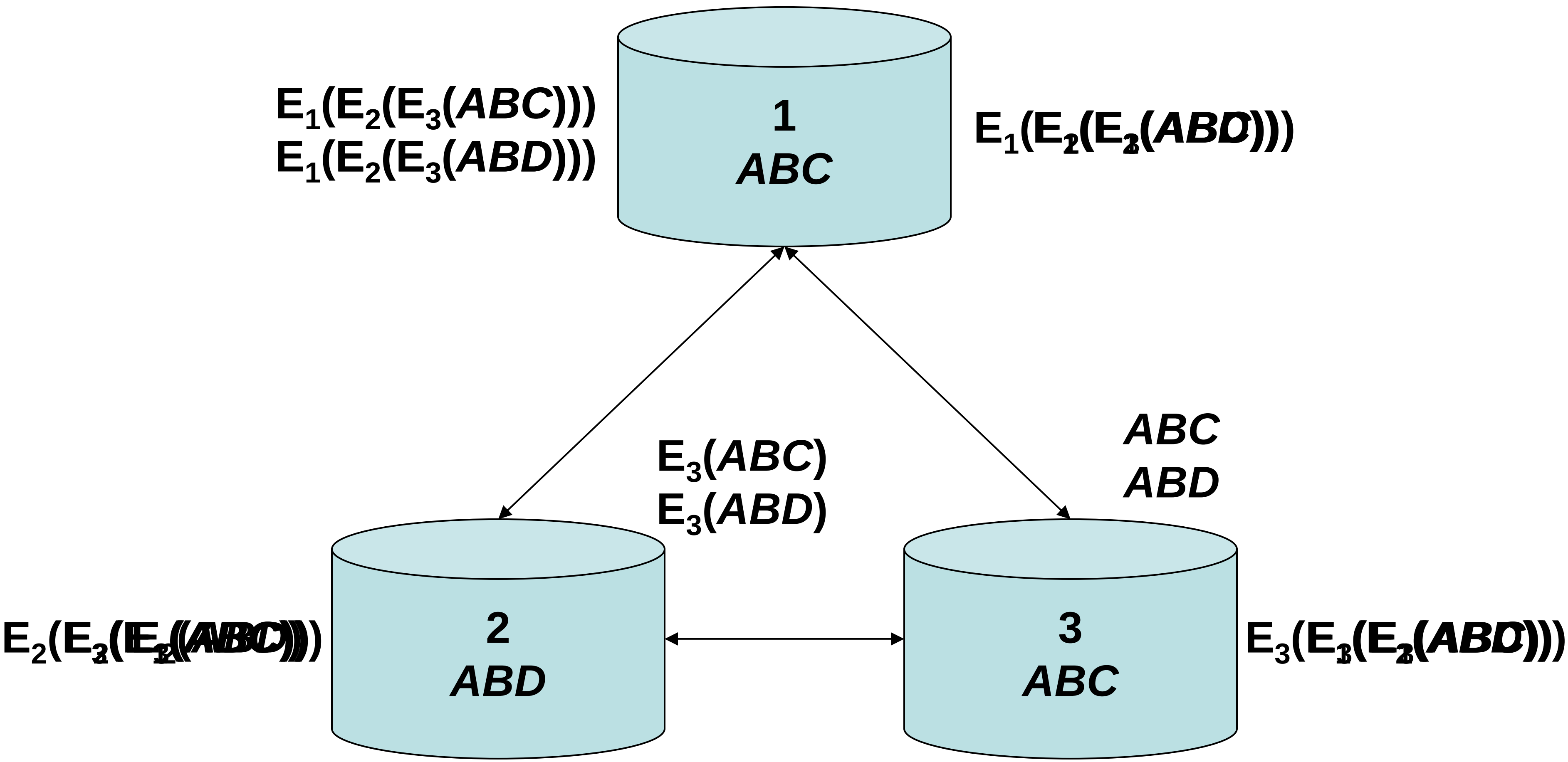
Securely Computing Candidates

- Key: Commutative Encryption ($E_a(E_b(x)) = E_b(E_a(x))$)
  - Compute local candidate set
  - Encrypt and send to next site
    - Continue until all sites have encrypted all rules
  - Eliminate duplicates
    - Commutative encryption ensures if rules the same, encrypted rules the same, regardless of order
  - Each site decrypts
    - After all sites have decrypted, rules left
- Care needed to avoid giving away information through ordering/etc.

Redundancy maybe added in order to increase the security.

Not fully secure according to definitions of secure multi-party

Computing Candidate Sets



Computing Globally Supported Itemsets

- Goal: To find globally supported large itemsets

$$X.\sup \geq s * \sum_{i=1}^{n} |DB_i|$$

$$\sum_{i=1}^{n} X.\sup_i \geq \sum_{i=1}^{n} s * |DB_i|$$

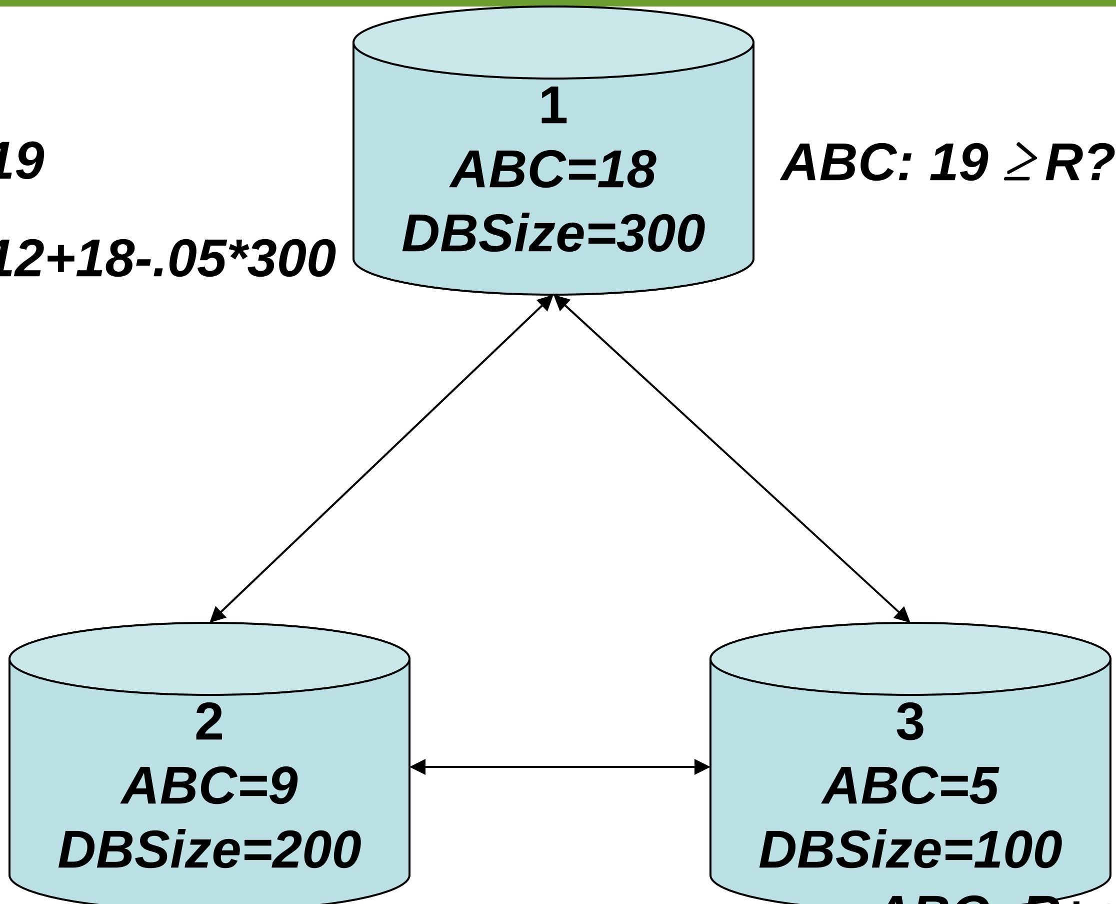
$$\sum_{i=1}^{n} \left(X.\sup_i - s * |DB_i| \right) \geq 0$$

Computing Frequent:
Is *ABC* ≥ 5%?

***ABC*: 19**

***ABC*: 12+18-.05*300**

***ABC*: 19 ≥ R?**



***ABC*: YES!**

R=17

***ABC*: R+count-freq.*DBSize**

***ABC*: 12**

***ABC*: 17+9-.05*200**

***ABC*: 17**

***ABC*: 17+5-.05*100**

Proof of Security

- We can simulate what is seen by each site by a simple random number generator. Because

$$\Pr\left[View_i^P = x\right] = \Pr\left[x_r = x - \sum_{k=1}^{k=i-1} x_k\right]$$

$$= \frac{1}{2^m}$$

$$= \Pr\left[\text{Simulator}_{\text{i}} = x\right]$$

- Therefore during addition nothing is revealed
- Assuming comparison is secure using secure composition thm., we are done.

Computing Confidence

- Checking confidence can be done by the previous protocol. Note that checking confidence for X $\Rightarrow$ Y

$$\frac{\{X \cup Y\}.\sup}{X.\sup} \geq c \Rightarrow \frac{\sum_{i=1}^{n} XY.\sup_i}{\sum_{i=1}^{n} X.\sup_i} \geq c$$

$$\Rightarrow \sum_{i=1}^{n} (XY.\sup_i - c * X.\sup_i) \geq 0$$

Secure Sub-protocols for PPDDM

- In general, PPDDM protocols depend on few common sub-protocols.

- Those common sub-protocols could be re-used to implement PPDDM protocols

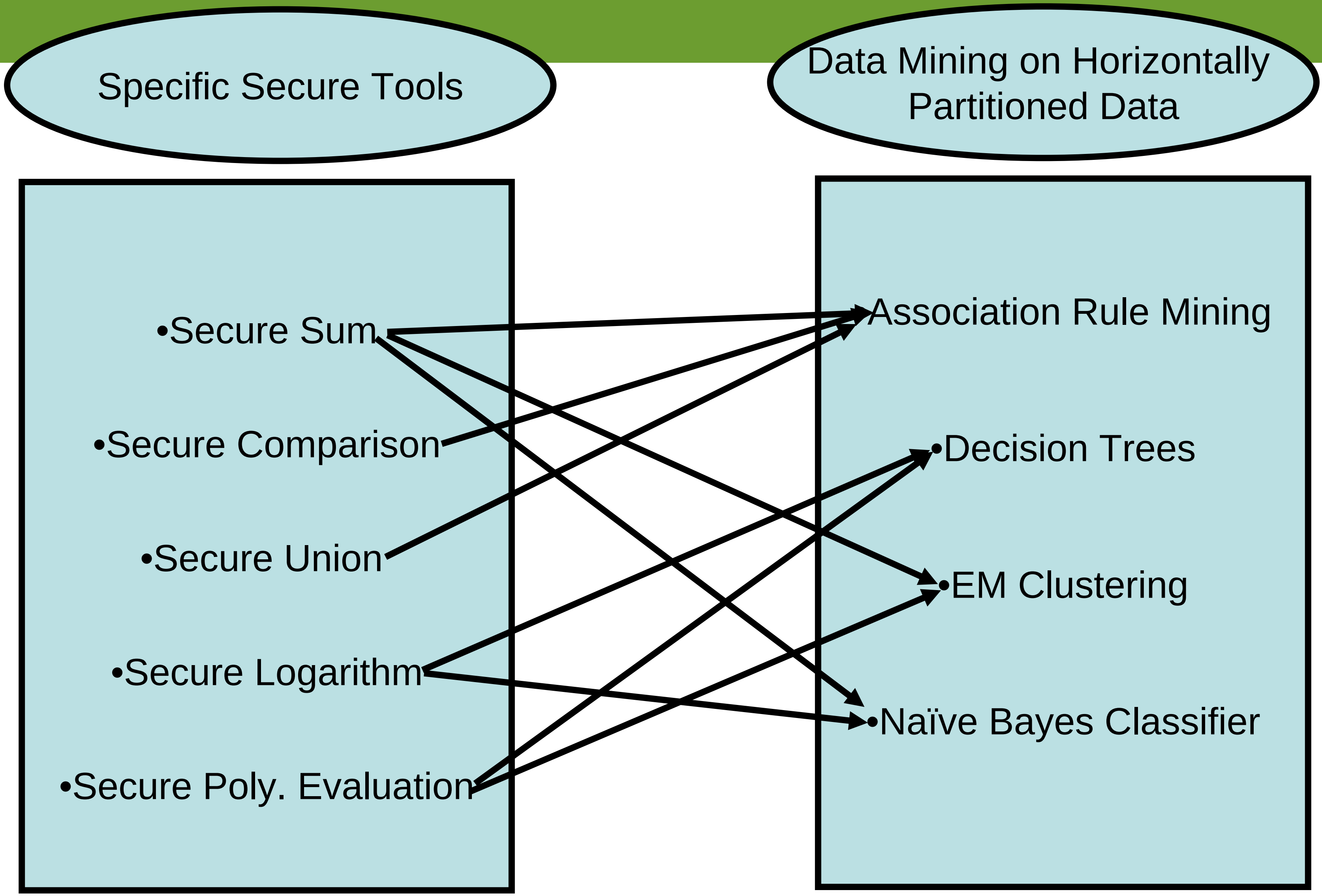
Secure Functionalities Used

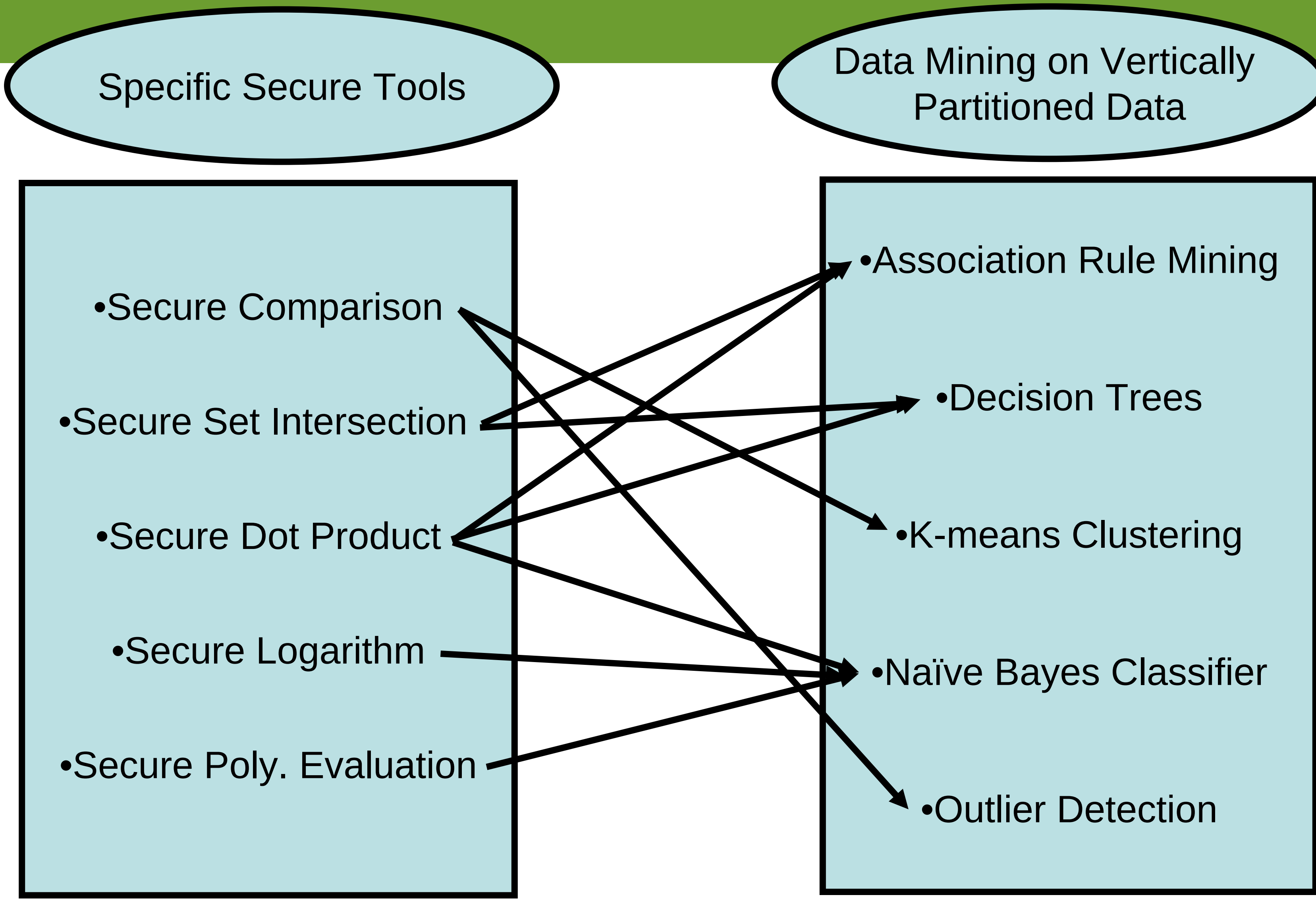
- **Secure Comparison**: Comparing two integers without revealing the integer values.
- **Secure Polynomial Evaluation**: Party A has polynomial P(x) and Part B has a value b, the goal is to calculate P(b) without revealing P(x) or b
- **Secure Set Intersection**: Party A has set S_A and Party B has set S_B , the goal is to calculate $S_A \cap S_B$ without revealing anything else.

Secure Functionalities Used

- **Secure Set Union:** Party A has set S_A and Party B has set S_B , the goal is to calculate $S_A \cup S_B$ without revealing anything else.

- **Secure Dot Product:** Party A has a vector X and Party B has a vector Y. The goal is to calculate X.Y without revealing anything else.


Specific Secure Tools
Data Mining on Horizontally Partitioned Data
•Secure Sum
•Secure Comparison
•Secure Union
•Secure Logarithm
•Secure Poly. Evaluation
Association Rule Mining
•Decision Trees
•EM Clustering
•Naïve Bayes Classifier


Specific Secure Tools
Data Mining on Vertically Partitioned Data
•Secure Comparison
•Secure Set Intersection
•Secure Dot Product
•Secure Logarithm
•Secure Poly. Evaluation
•Association Rule Mining
•Decision Trees
•K-means Clustering
•Naïve Bayes Classifier
•Outlier Detection

Summary of SMC Based PPDDM

- Mainly used for distributed data mining.
- Provably secure under some assumptions.
- Learned models are accurate
- Efficient/specific cryptographic solutions for many distributed data mining problems are developed.
- Mainly semi-honest assumption(i.e. parties follow the protocols)
- Malicious model is also explored recently.
- Many SMC based PPDM algorithms share common sub-protocols (e.g. dot product, summation, etc.)

Drawbacks for SMC Based PPDDM

- Drawbacks:
  – Still not efficient enough for very large datasets. (e.g. petabyte sized datasets ??)
  – Semi-honest model may not be realistic
  – Malicious model is even slower